IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number:	10/734,966
Filing Date:	12/15/2003
Applicant(s):	Mustansir Banatwala, et al.
Entitled:	EVENT NOTIFICATION STRUCTURE FOR DYNAMICALLY AGGREGATED LOGICAL COMPONENTS
Examiner:	KimbleAnn C. Verdi
Group Art Unit:	2194
Attorney Docket No.:	LOT9-2003-0078US1 (7321-032U)

TRANSMITTAL OF APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Submitted herewith is Appellant's Appeal Brief in support of the Notice of Appeal filed March 24, 2008. This Appeal Brief has been timely filed within the statutory period to provide an Appeal Brief from the date of the Notice of Appeal and any required fees for an extension of time under 37 C.F.R. § 1.136 are provided herewith. Notwithstanding, please charge any shortage in fees due under 37 C.F.R. §§ 1.17, 41.20, and in connection with the filing of this paper to Deposit Account 12-2158, and please credit any excess fees to such deposit account.

Date: May 29, 2008

Respectfully submitted,

/Steven M. Greenberg/
Steven M. Greenberg, Registration No. 44,725
Customer Number 46321

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number:	10/734,966
Filing Date:	12/15/2003
Applicant(s):	Mustansir Banatwala, et al.
Entitled:	EVENT NOTIFICATION STRUCTURE FOR DYNAMICALLY AGGREGATED LOGICAL COMPONENTS
Examiner:	KimbleAnn C. Verdi
Group Art Unit:	2194
Attorney Docket No.:	LOT9-2003-0078US1 (7321-032U)

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed March 24, 2008, wherein Appellants appeal from the Examiner's rejection of claims 1, 3 through 6 and 8 through 18.

I. REAL PARTY IN INTEREST

This application is assigned to International Business Machines Corporation by assignment recorded on December 15, 2003, at Reel 014802, Frame 0357.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1, 3 through 6 and 8 through 18 are pending in this Application and have been twice rejected, claims 2 and 7 having been canceled in the Amendment dated October 30, 2007 (the "Amendment"). It is from the multiple rejections of claims 1, 3 through 6 and 8 through 18 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

Claims 1, 3, 5, 8, 9, 10, 11, 12, 14, 15, 16 and 17 were amended once in the Amendment in response to the Non-Final Office Action dated July 30, 2007 (the "Non-Final Office Action") and claims 2 and 7 were canceled in the Amendment.

V. SUMMARY OF CLAIMED SUBJECT MATTER

By reference to paragraph [0018] of Appellants' published specification, independent claims 1, 6, 9 and 14 are respectively directed to an event notification and management system, a dynamic proxy configured for interoperation with a component instance in a dynamic aggregation of components, an event notification and management method and a machine readable storage having stored thereon a computer program for event notification and management. In accordance with Appellants' invention, logical components in an aggregation can be configured at the time of creation with respective instances of a dynamic proxy. Each dynamic proxy can trap events of interest in a coupled logical component. Once trapped, the events of interest can be posted to an event broker which can publish the events to registered

subscribers. Preferably, in addition to providing event notification, the dynamic proxy can manage the event notification which management can range from quashing the event to fully handling the event. In all cases, a list of events published and managed through the dynamic proxy can be defined by the registered subscribers in order to limit resource consumption in the event notification framework. In this way, different computing processes within an aggregation can have an awareness of one another to permit event notification and management.

With specific reference to claim 1, an event notification and management system can include aggregation of logical components, each logical component having a coupling to a corresponding dynamic proxy. (Par. [0020]) The system also can include an event notification service communicatively linked to subscribing processes, and an event queue disposed between the dynamic proxy and the event notification service. (Par. [0024]) Notably, the dynamic proxy can include event notification logic and event management logic. (Par. [0023])

With specific reference to claim 6, a dynamic proxy can be configured for interoperation with a component instance in a dynamic aggregation of components. (Par. [0023]) The dynamic proxy can include a list of selected listener method calls in the component instance and can be communicatively coupled to an event queue. (Par. [0023]) The dynamic proxy also can include event notification logic coupled to the list and configured to post events to the event queue which relate to invoked listener method calls included in the list. (Par. [0024]) Finally, the dynamic proxy can include event management logic configured to selectively handle invoked listener method calls. (Par. [0027])

With specific reference to claim 9, an event notification and management method can include creating a component instance from an amalgamation of a dynamic proxy object definition and a component interface. (Par. [0020]) The method also can include trapping selected calls to methods disposed within the component instance, and routing a reference to the trapped selected calls to an event queue. (Par. [0025]) Finally, for each reference in the event queue, a notification can be distributed to a set of subscribers registered to receive notifications for the reference. (Par. [0025])

With specific reference to claim 14, a machine readable storage can have stored thereon a computer program for event notification and management. The computer program can include a routine set of instructions which when executed by a machine cause the machine to create a component instance from an amalgamation of a dynamic proxy object definition and a component interface. (Par. [0020]) The computer program also can include a routine set of instructions which when executed by the machine cause the machine to trap selected calls to methods disposed within the component instance and to route a reference to the trapped selected calls to an event queue. (Par. [0025]) Finally, the computer program can include a routine set of instructions which when executed by the machine cause the machine to distribute a notification to a set of subscribers registered to receive notifications for each reference in the event queue. (Par. [0025])

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 14 through 18 have been rejected under 35 U.S.C. § 101 as reciting non-statutory subject matter.

Claims 1 through 7, 9 through 12, and 14 through 17 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Harrison et al., The Design and Performance of a Real-Time CORBA Event Service, in OOPSLA 1997 (ACM 1997), hereinafter "Harrison" in view of United States Patent No. 6,269,396 to Shah et al. (Shah).

Claims 8, 13, and 18 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Harrison in view of Shah and further in view of United States Patent No. 6,877,163 to Jones et al. (Jones).

VII. THE ARGUMENT

THE REJECTION OF CLAIMS 14 THROUGH 18 UNDER 35 U.S.C. § 101.

For convenience of the Honorable Board in addressing the rejections under 35 U.S.C. § 101, claims 15 through 18, stand or fall together with independent claim 14.

Appellants' Claim 14 recites a machine readable storage having stored thereon a computer program for event notification and management. Specifically, claim 14 reads:

> 14. A machine readable storage having stored thereon a computer program for event notification and management, the computer program comprising a routine set of instructions which when executed by a machine cause the machine to perform the steps of:
>
> creating a component instance from an amalgamation of a dynamic proxy object definition and a component interface;
>
> trapping selected calls to methods disposed within said component instance;
>
> routing a reference to said trapped selected calls to an event queue; and,
>
> for each reference in said event queue, distributing a notification to a set of subscribers registered to receive notifications for said reference.

As part of claim 14, several physically transformative steps are performed. First, a component instance is created from an amalgamation of a dynamic proxy object definition and a component interface. Second, selected calls to methods disposed within the component instance are trapped

and a reference to the trapped selected calls is routed to an event queue. Finally, for each reference in the event queue, a notification is distributed to a set of subscribers registered to receive notifications for the reference.

In the Final Office Action, Examiner persists in asserting a rejection under 35 U.S.C. 101 in respect to claims 14 through 18 because, in Examiner's view, the subject matter recited by claims 14 through 18 is directed to non-statutory subject matter. Specifically, Examiner's entire argument is reproduced herein for the convenience of the Honorable Board.

> With respect to claims 14-18, a "machine readable storage having stored thereon a computer program for event notification and management, the computer program comprising" is being recited; however, it appears that machine readable storage having stored thereon a computer program for event notification and management, the computer program comprising would reasonably be interpreted by one of ordinary skill in the art as software, per se. A machine readable storage having stored thereon a computer program for event notification and management, the computer program comprising as claimed does not set forth a means to realize the software, per se such as being stored in a memory or computer storage media. As such, it is believed that a machine readable storage having stored thereon a computer program for event notification and management, the computer program comprising of claims 14-18 is reasonably interpreted as functional descriptive material, per se.

Appellants previously addressed Examiner's rejections by noting that claims 14-18 specifically recite "a computer readable storage" and that a computer usable/readable storage is an article of manufacture and, thus, is statutory. In this regard, Appellants directed Examiner to M.P.E.P. § 2106.01, which states:

> When functional descriptive material is recorded on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized.

Appellants also observed that one skilled in the art would recognize that machine-readable storage includes memory, hard drives, floppy drives, USB keys, etc and that a quick search of the

U.S.P.T.O patent database revealed literally tens of hundreds of issued patents having claims with the term "computer readable storage".

In response, Examiner stated, "[E]xaminer respectfully disagrees and notes that "computer readable storage" is not recited in the rejected claim(s)." Examiner further argued, "In addition claim 14 as written "A machine readable storage having stored thereon a computer program for event notification and management, the computer program comprising" appears to claim a computer program not an article of manufacture." Yet, Examiner provides no legal foundation as to how a "machine readable storage" and a "computer readable storage" differ in respect to an analysis under 35 U.S.C. § 101 and how any such perceived distinction results in "software per se" whereas Examiner appears to concede that a "computer readable storage" is not "software per se". Finally, Examiner objects to the term "machine readable storage having stored thereon a computer program" as lacking antecedent basis in the specification, notwithstanding the clear teachings set forth in paragraph [0033] of Appellants' specification.

In any event, Appellants can find no reference in the Manual of Patent Examining Procedure (M.P.E.P.) to "software per se", though it is important to recognize portions of the M.P.E.P. which are dispositive in determining statutory subject matter. First, M.P.E.P. 2106(IV)(C)(2) sets forth

> A claimed invention is directed to a practical application of a 35 U.S.C. 101 judicial exception when it:
> (A) "transforms" an article or physical object to a different state or thing; or
> (B) otherwise produces a useful, concrete and tangible result, based on the factors discussed below.

Of substantial importance, M.P.E.P. 2106(IV)(C)(2)(1) provides in pertinent part, "USPTO personnel first shall review the claim and determine if it provides a transformation or reduction of an article to a different state or thing. If USPTO personnel find such a transformation or

reduction, USPTO personnel **shall end the inquiry and find that the claim meets the statutory requirement of 35 U.S.C. 101**." Appellants have clearly established multiple different transformations of an article to a different state or thing--namely the creation of a component instance, the trapping of selected calls to methods disposed within the component instance and the routing of the trapped selected calls to an event queue, to name only a few transformations.

Examiner may be tempted to refer to M.P.E.P. 2601.01 in persisting in a "software per se" argument, however, Appellants also reference section I of M.P.E.P. 2601.01 in which it is stated in relevant part:

> Computer programs are often recited as part of a claim. USPTO personnel should determine whether the computer program is being claimed as part of an otherwise statutory manufacture or machine. In such a case, the claim remains statutory irrespective of the fact that a computer program is included in the claim. **The same result occurs when a computer program is used in a computerized process where the computer executes the instructions set forth in the computer program.** Only when the claimed invention taken as a whole is directed to a mere program listing, i.e., to only its description or expression, is it descriptive material per se and hence nonstatutory.

In claim 14, a machine readable storage is recited that has stored thereon a computer program for event notification and management. As recited in claim 14, the computer program includes a routine set of instructions which when executed by a machine cause the machine to perform the enumerated steps. Accordingly, by all accounts, claim 14 is statutory under M.P.E.P. 2106.01(I).

THE REJECTION OF CLAIMS 1 THROUGH 7, 9 THROUGH 12 AND 14 THROUGH 17 UNDER 35 U.S.C. § 103(A)

For convenience of the Honorable Board in addressing the rejections under 35 U.S.C. § 103(a), claims 3 through 5 stand or fall together with independent claim 1, claims 8 through 13 stand or fall together with independent claim 6, and claims 15 through 18 stand or fall together with independent claim 14. As amended claim 1 recites

> 1. An event notification and management system comprising:
> an aggregation of logical components, each logical component having a coupling to a corresponding dynamic proxy;
> an event notification service communicatively linked to a plurality of subscribing processes; and, an event queue disposed between said dynamic proxy and said event notification service;
> **wherein at least one of said dynamic proxy comprises event notification logic and event management logic.**

Initially in the Non-Final Office Action, Examiner asserted only Harrison in support of a rejection of claims 1 and 6 under 35 U.S.C. § 102, however, in the Amendment dated October 30, 2007, Appellants argued that Harrison did not teach both event notification logic and event management logic present in the dynamic proxy.

Examiner appears to agree and in the Final Office Action, Examiner now refers to the combination of Harrison and Shah in support of the rejections under 35 U.S.C. § 103(a). Specifically, Examiner states in the Final Office Action,

> Harrison does not explicitly disclose wherein at least one of said dynamic proxy comprises event notification logic and event management logic. However [Shah] teaches wherein at least one of said dynamic proxy comprises event notification logic and event management logic (col. 26, lines 58-67).

The entirety of the cited portion of Shah, however, provides no teaching directed to the inclusion of both event notification logic and event management logic in a dynamic proxy. The cited portion of Shah is reproduced herein for the convenience of the Honorable Board.

> The event manager subsystem provides the ability for a users to generically issue event notification to one or more registered parties. Multiple Event::Manager object instances may exist in the system. A node level Event::Manager exists on all nodes. Other Event::Manager instances may also exist to provide the ability for interested parties to register for events that are special to a process. The eventmanagerimpl program provides an Event::Manager object instance for the mode that it is running on. Events that are relevant to a node get issued through that Event::Manager instance. Users interested in events on a particular node can bind to that nodes Event::Manager instance by using that nodes name as the Event::Manager name. Programs can also embed an Event::Manager object within their program.

Thus, it appears Examiner has merely "keyword" searched Shah and upon finding the term "event manager", Examiner has applied Shah without considering its teachings in context. To wit, the word "proxy" is wholly absent from the forty-one (41) pages of Shah. Further, the notion of "event notification logic" not only cannot be found in the cited portion of Shah, but Examiner has made no attempt to argue so.

THE REJECTION OF CLAIMS 8, 13, AND 18 UNDER 35 U.S.C. § 103(A)

In that the combination of Harrison and Shah fail to show both event management logic and event notification logic within a dynamic proxy, at least on this basis alone the Examiner's rejections of claims 8, 13 and 18 cannot stand.

Based upon the foregoing, Appellant respectfully submit that the Examiner's rejections under 35 U.S.C. §§ 101 and 103(a) based upon the applied prior art are not viable. Appellants, therefore, respectfully solicit the Honorable Board to reverse the Examiner's rejections under 35 U.S.C. §§ 101 and 103(a).

Date: May 29, 2008

Respectfully submitted,

/Steven M. Greenberg/
Steven M. Greenberg
Registration No. 44,725
Customer Number 46321

VIII. CLAIMS APPENDIX

1. (Previously Amended) An event notification and management system comprising:

an aggregation of logical components, each logical component having a coupling to a corresponding dynamic proxy;

an event notification service communicatively linked to a plurality of subscribing processes; and, an event queue disposed between said dynamic proxy and said event notification service;

wherein at least one of said dynamic proxy comprises event notification logic and event management logic.

2. Canceled

3. (Previously Amended) The system of claim 1, further comprising an event list coupled to at least one of said dynamic proxy.

4. (Original) The system of claim 1, further comprising an event-to-subscriber list coupled to said event notification service.

5. (Previously Amended) The system of claim 1, further comprising an event action list coupled to at least one of said dynamic proxy.

6. (Original) A dynamic proxy configured for interoperation with a component instance in a dynamic aggregation of components, the dynamic proxy comprising:

a list of selected listener method calls in said component instance;

a communicatively coupling to an event queue;

event notification logic coupled to said list and configured to post events to said event queue which relate to invoked listener method calls included in said list; and,

event management logic configured to selectively handle invoked listener method calls.

7. Canceled

8. (Previously Amended) The dynamic proxy of claim 7, wherein said event management logic comprises programming for selectively performing one of quashing an one of said invoked listener method calls, handling said invoked one of said listener method calls without passing said one of said invoked listener method calls to the component instance, assisting the component instance in handling said one of said invoked listener method calls while passing said one of said invoked listener method calls to the component instance, and modifying said one of said invoked listener method calls before passing said one of said invoked listener method calls to the component instance.

9. (Previously Amended) An event notification and management method comprising the steps of:

creating a component instance from an amalgamation of a dynamic proxy object definition and a component interface;

trapping selected calls to methods disposed within said component instance; routing a reference to said trapped selected calls to an event queue; and,

for each reference in said event queue, distributing a notification to a set of subscribers registered to receive notifications for said reference.

10. (Previously Amended) The method of claim 9, wherein said creating a component instance comprises the step of instructing a factory object coupled to said dynamic proxy object definition to create said component instance.

11. (Previously Amended) The method of claim 9, further comprising the steps of:

determining whether an event notification service responsible for said distributing a notification has been activated; and,

performing said routing a reference only if said event notification service has been activated.

12. (Previously Amended) The method of claim 9, wherein said routing a reference comprises the steps of:

consulting an event list enumerating specific ones of said trapped selected calls; and,

posting to said event queue only specific ones of said calls included in said event list.

13. (Original) The method of claim 9, further comprising the step of selectively performing one of quashing said trapped selected calls, handling said trapped selected calls without passing said trapped selected calls to said component instance, assisting said component instance in handling said trapped selected calls while passing said trapped selected calls to said component

instance, and modifying said trapped selected calls before passing said trapped selected calls to said component instance.

14. (Previously Amended) A machine readable storage having stored thereon a computer program for event notification and management, the computer program comprising a routine set of instructions which when executed by a machine cause the machine to perform the steps of:

creating a component instance from an amalgamation of a dynamic proxy object definition and a component interface;

trapping selected calls to methods disposed within said component instance;

routing a reference to said trapped selected calls to an event queue; and,

for each reference in said event queue, distributing a notification to a set of subscribers registered to receive notifications for said reference.

15. (Previously Amended) The machine readable storage of claim 14, wherein said creating a component instance comprises the step of instructing a factory object coupled to said dynamic proxy object definition to create said component instance.

16. (Previously Amended) The machine readable storage of claim 14, further comprising the steps of:

determining whether an event notification service responsible for said distributing a notification has been activated; and,

performing said routing a reference only if said event notification service has been activated.

17. (Previously Amended) The machine readable storage of claim 14, wherein said routing a reference comprises the steps of:

consulting an event list enumerating specific ones of said trapped selected calls; and,

posting to said event queue only specific ones of said calls included in said event list.

18. (Original) The machine readable storage of claim 14, further comprising the step of selectively performing one of quashing said trapped selected calls, handling said trapped selected calls without passing said trapped selected calls to said component instance, assisting said component instance in handling said trapped selected calls while passing said trapped selected calls to said component instance, and modifying said trapped selected calls before passing said trapped selected calls to said component instance.

IX. EVIDENCE APPENDIX

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 of this title or of any other evidence entered by the Examiner has been relied upon by Appellant in this Appeal, and thus no evidence is attached hereto.

X. RELATED PROCEEDINGS APPENDIX

Since Appellant is unaware of any related appeals and interferences, no decision rendered by a court or the Board is attached hereto.